

Complement And Add (MK14) Surviving Issues Written by Geoff Phillips 1979-1980.

Preface 2019

“Complement and Add” was an instruction of the SC/MP microprocessor, the CPU in the MK14, one of the earliest affordable home computers that one could buy in the UK, and other parts of Europe, in 1978. It’s actually a long winded way of saying “subtract” (with carry).

When I bought an MK14 in 1978 (I think), I remember it took a long time to arrive – delay after delay, then finally it arrived, various letters of apology from Science of Cambridge. Then I spent hours constructing it from its kit form. Not the best at soldering, I had a few problems to start with down to my poor soldering, but soon I was up and running and ready to use its 256 byte RAM to write some programs. (You could add an optional 256 bytes extra)

1978 is a kind of prehistory in terms of microcomputers. Americans were of course way ahead of the game, but in the UK home computing was strongly allied to electronics, and so there were things like a month-by-month construction of a home computer if you had the ability and patience (and money) to make PCBs and solder your own modules together. There were enormously expensive machines one could buy for the equivalent of several thousand pounds in today’s money. It was a time when people simply did not know what a home computer might do, but I think we all had great hopes. We knew we needed much more memory, storage, and video output with a high a screen resolution as possible.

I had been a computer programmer professionally since 1977, writing COBOL programs for John Laing, the building company. A programmer in this era did not see much of the computer – there was virtually no hands-on. Jobs were submitted through punched cards, or coded forms handed over to be keyed in and the output returned some hours later.

So the MK14 plunged you into the deep end, because from the start you had to write machine language instructions, and enter them in Hex. The limited operating system merely gave you the ability to set an address pointer, to input bytes, to view memory again as bytes, and then to run programs. The original board had no way to save your code, but since programs tended to be very short, this was not a huge disadvantage. The MK14 would seem slow by today’s standards of course. But at the time, the only equivalent programmable machine I owned was in the form of programmable calculators, and I had owned a few of those.

The MK14 came with a slender manual which did have a number of simple programs and games to give you an example to work from. I think any owners of the MK14 probably read that little booklet backwards and forwards trying to get as much sense as possible out of it for clues as to what to do.

Back in these early days when home computing, such as it was, was part and parcel of hobby electronics there was not a great deal of help out there to explain any thing about the methods of writing any useful applications. For some months, I worked

alone, as I suppose many others did. Books about home computers were scarce. I would often go to Foyles bookshop in London and at this time there would be perhaps 4 large book cabinets devoted to computers, with some crossover in the larger electronics section. And this, being a bookshop, meant multiple copies of the same book. Most were about long established subjects, Cobol, Fortran, a few about BASIC of different dialects. What a difference a few years made – by 1983 Foyles had a whole wing set for computers, which I remember was always very busy.

The SC/MP was a great introductory microprocessor. Its limitations were so severe as to make the following processors like the Z80 and 6502 absolutely luxurious. One thing I had forgotten (erased from my memory) was that the SC/MP had relative addressing. This had one advantage that code would work mostly even if loaded to a different address, but required an enormous effort in hand coding everything, and hand coding was the only thing available. So for example, if you had an instruction at 0B00 which wanted to load from memory at 0B60 you would calculate the byte distance from your current instruction to 0B60. Move the instruction up say, to insert an instruction, and the offsets all change – horrible looking back now. But it also had 3 additional pointers, so you could point at a bit of RAM and use variables that way – my memory is hazy now of how I did things.

On the 6502 there was the wonderful zero page idea, there, a location like loading from 60 was always just that, 60, so with the economy of not needing the high part of the address. The SC/MP also had no stack, unless you implemented one yourself. This really is part and parcel of the idea of a small memory processor, one not really suited to large scale projects. Reading the newsletters, it becomes apparent that I would have liked a larger scale version of the SC/MP processor, but once I started programming 6502 and Z80, the convenience of their respective architectures was immediately obvious. Also, what became standard during the 1980's was the idea of connecting to a television, tuning a spare channel to the modulator output. Cassette tape gave way to diskette drives, and I don't think we really looked back. I remember there was a machine even earlier than the MK14 which was entirely controlled by single bit switches, and corresponding single bit LEDs. When I became a support programmer at Laings, I would come in when a program crashed overnight, and the rules were relaxed a little, so I recall seeing the IBM 370 mainframe close up. I remember seeing that it had binary lights showing the address and data bus. During my years there, it was replaced by something "compatible" which worked four times faster. All the overnight runs that would finish at 4 in the morning were all complete by 10pm. There would also be online terminals, and so the heavy-going punched cards and coding forms regime gave way to the kind of way everyone uses computers today.

So back to the MK14. When my MK14 was built, I spent a long time trying out the example programs, and learning how to write machine language code. After a while, the hex codes of the instruction set fused into my head and this is why often code listings in the newsletters are of just raw hex, remember we did not have the convenience of assemblers, and even with a cassette interface to save code, everything took time.

A new magazine was launched – Personal Computer World, and I wrote a letter, published in the second issue, asking if anyone was interested in a user group. The

magazine said something like “this is a great idea, we’re sending you £5.00”. (This never arrived by the way)

What did arrive was a bundle of steady mail over the following month, and surprisingly from all over the world. One person lived nearby and became a friend, and later I worked for him, and it’s true to say that one letter changed my life. So I started a user club, initially free (“but please send stamps”) and later for the somewhat low price of £1.00 a year which sounds ridiculous now. I wanted to run it non profit – of course it actually cost me a lot to run, and people would, by my request, sometimes pay in stamps. And I started to create “Complement And Add”, a pun on the SC/MP instruction, a small hand-typed newsletter with programs and contributions. I used carbon paper, so wrote out a rough draft then carefully typed several copies at once. The first issue is dated February/March 1979. The idea was that one person would send to the next in the list, and add anything of their own, until it came back to me, and anything worth having would get printed in a subsequent newsletter, and the odd loose scraps of paper from other people might be sent too.

Well, perhaps a naïve idea, because of course what happened is people did not bother sending on the publication, and then people did not get them. Eventually, I just had a number of them photocopied, and sent them to anyone who complained. It was all a bit messy. In my files, there are two or three hundred letters. I always replied, and often spent a long time answering questions and giving technical advice.

I had an odd kind of non-relationship with Science of Cambridge. In letters from people they would sometimes say they had been told by S of C of my user club and they were notoriously bad at supporting their customers. But you might think that just once they might call and say thanks for being there. I once had a phone call from Chris Curry while I was at work asking me to go and pacify an irate customer on the other side of London to me, who was threatening legal action because his MK14 had some kind of a fault. I did not drive at this time, and took a day off, and worked my way to far SE London (St Mary’s Cray) to find this address. Once he realised I was not working for S of C directly, he was actually very nice, and he tried to show me the problem (the details of which I forget). But nothing went wrong, with that peculiarity that machines have of not going wrong when they are being watched, the MK14 stubbornly kept running for two hours. I sent a report and expenses to S of C – never heard a thing back nor received the expenses (which were some absurdly low amount)

I created a newsletter about every two months, but it was a volatile time for home computers, and after just a year, about March 1980 the club had reached the end of its useful life. My own interest was waning, and I think realistically too that of club members. Sadly the later software does not survive at all, in any form. I remember doing a great space invaders which had a bug which created an effect that I liked, sparkling bombs, or something.

1979-1982 was an amazingly volatile time for home computers. Before things settled a bit with the ZX80, then the ZX81 and Spectrum; the early Atari and Commodore computers and a few others, there was a huge variety of small computer manufacturers who were trying their hand at home computers. So the MK14 was soon outdone by home computers with better features, more memory (you could not really have less!) and output to the telly.

So by 1980, I needed to move on and learn 6502, the processor used in a good many early computers, not least of which was the influential BBC Micro. So the newsletters stopped. In all, there were perhaps eight or nine issues. Years later, I created Complement and Add 2001 (2001 was still far in the future) for a joke, and sent it to my main correspondents who probably thought I was mad. I am afraid I no longer have this. In my file of letters, the MK14 correspondence dries up after about August 1980.

Not all parts of the newsletters survive, and sadly not many additional programs. But my initial thoughts were that none had survived at all! By 1995 when I moved out of my office in Ely, I had accumulated 15 years of old stuff. My cassette tapes had been reused for the other tape based machines, and then later for music. Little additional software survives, and I know I wrote a huge amount. Because of the unreliability of the tape interface there were projects I lost even while still writing code for the MK14.

One newsletter was printed on the electrostatic paper of the first printer I owned - a terrible idea, because it's so narrow - and nothing of that remains. There is also a problem in that I sometimes retyped issues, and in retyping, changed the contents slightly. What survives of issue one is a mish-mash of different copies. I am also not sure about certain pages belonging to certain issues so have just done my best. I've assembled various extra bits to create an eighth issue. Also in reading these, I find that the writing is very self-indulgent rather than matter of fact, but it is what it is. Almost all of these documents are here in their original form.

Later on, and really too late realistically, since people had moved on to better machines - someone approached me about writing an operating system that was based on the VDU card. Somewhat stupidly, I agreed, wasting months on the project that was never published or used - it sat on Eproms. It is perhaps something that we would all have loved when the MK14 was first sold - a few K of RAM, and inbuilt inline assembler and output onto television. My MK14 in the end was in an inverted record deck top, with straggling wires everywhere, and extra RAM somewhat haphazardly added, with the video and cassette board. Actually, this was at least my second one, the first one packed in, and they started selling prebuilt units, or possibly another company did. I suppose it was all binned in the early 1980's when there were a million other things going on.

The MK14 was a very good thing for me - I made two good friends out of it, one would lead to writing two books, about the Dragon 32 and the Oric-1. The other led to working for Tansoft, for various Oric projects, and then that led onto other things, and my whole life's work experience was a sequence that started from that crazy path.

So it's now 40 years later, and so much has happened. I left John Laing's in 1983 to become a games programmer. I wrote programs for the Dragon 32, a UK101, a Microtan, an Oric-1, a Commodore 64, MSX, Amstrad, Atari ST, Amiga and of course finally the PC. I kept in touch for many years with certain people who were contributing articles and programs. After the ups and downs of the games programming, I went to work for MSU who were making the Konix, which was a games machine. Here, I was writing system software rather than the games themselves. The ASIC was reused for non-games applications, like video players and

set top internet boxes: and working here was a very happy time. I am still, in 2019, a programmer, but working part time for a small TV studio in Milton Keynes. I enjoy coding, and work on things privately that interest me.

The last issue of C&A that I have has, in a roundabout way, quite a prediction at the end of my short story where I am speaking ironically – and remember this was 1980 – “... It’s problems like this which are prevented by the marvellous idea of having different: MPU’s/Keyboard layouts/Bus structures/cassette interface formats/Dialect of languages.. After all we don’t want these machines to actually start communicating, do we?” Today, computers are all about communication, so somewhere along the line we did learn our lesson, at least partially. I’ve obliterated any personal addresses for obvious reasons.

I’ve OCR’d the C&As and checked the code where I can. But I also include the images too in most cases.

I hope these “Complement and Add” newsletters are of interest.

Geoff Phillips